

ICS: 33.160

CCS: M61



UHD World Association  
世界超高清视频产业联盟

# 世界超高清视频产业联盟标准

T/UWA 005.2-1-2022

---

## 高动态范围（HDR）视频技术

### 第 2-1 部分：应用指南 系统集成

High Dynamic Range Video Technology

Part 2-1: Application Guide System Integration

（ V1.1 版）

2022-02-10 发布

2022-02-10 实施

---

世界超高清视频产业联盟



## 目 录

前 言.....	II
1. 范围.....	1
2. 规范性引用文件.....	1
3. 术语和定义.....	1
3.1 HDR Vivid.....	1
3.2 HDR Vivid 播放设备.....	1
3.3 HDR Vivid 显示设备.....	1
4. 缩略语.....	1
5. 概述.....	2
6. HDR Vivid 的 ES 流封装.....	2
6.1. 总体要求.....	2
6.2. HEVC/H.265 以及 VVC/H.266 ES 流封装.....	2
6.3. AVS2 以及 AVS3 ES 流封装.....	3
6.4. 版本后向兼容.....	6
7. HDR Vivid MP4 文件封装.....	6
8. HDR Vivid HLS 流封装.....	8
8.1. 总体要求.....	8
8.2. HLS 流封装.....	8
9. HDR Vivid DASH 封装.....	9
9.1. 总体要求.....	9
9.2. DASH 流封装.....	9
10. HDR Vivid DVB TS 流封装要求.....	9
10.1. 总体要求.....	9
10.2. TS 流的封装.....	9
10.3. TS 流的丢包处理.....	11
11. 终端应用说明.....	11
11.1. 终端系统组成.....	11
11.2. HDR Vivid 信号终端解码与呈现.....	11
11.3. 动态元数据 HDMI 传输要求.....	13
11.3.4. 接收端适配模式显示端适配过程的图形处理.....	19
12. 后期制作的应用说明.....	21
12.1. 输入输出建议.....	21
12.2. 画面质量监控.....	21
12.3. 动态元数据生成模式.....	21
13. 内容保护.....	24

## 前 言

本标准按照 GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准主要起草单位：杭州当虹科技股份有限公司、深圳市腾讯计算机系统有限公司、成都索贝数码科技股份有限公司、深圳赛西技术股份有限公司、华为技术有限公司、上海海思技术有限公司、中国电子技术标准化研究院、中央广播电视总台、国家广播电视总局广播电视规划院、国家广播电视总局广播电视科学研究院、北京爱奇艺科技有限公司、字节跳动网络技术有限公司、上海哔哩哔哩科技有限公司、北京天工异彩影视科技有限公司、天津锐彩智作影视科技有限公司、北京朱氏兄弟文化传播有限公司、迈思非墨（北京）文化传媒有限公司、北京上添影视传媒有限公司、中兴通讯股份有限公司、北京数码视讯科技股份有限公司、京东方科技集团股份有限公司、中国移动通信集团有限公司、晶晨半导体（上海）股份有限公司、中国电信集团有限公司。

本标准主要起草人：陈勇、朱正元、王建伟、张金沙、张旭、李育中、汪亮、张旭、王江涛、余全合、陈虎、袁乐、黄芳、李岩、王佩、宁金辉、郭晓强、王志航、李勇鹏、符乐安、黄传增、王一、叶天晓、郑龙、周辉、王博、朱永华、丁业恒、张亘、陈仁伟、周聘、顿胜堡、郭佩佩、杨忠尧、陈月、罗传飞。

# 高动态范围（HDR）视频技术第 2-1 部分： 应用指南 系统集成

## 1. 范围

本文件规定了支持 T/UWA 005.1-2022 规定的 HDR 系统集成总体内容，包括制作，传输和接收和解码。

## 2. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改版本）适用于本文件。

ISO/IEC 14496-12:2020 Information technology — Coding of audio-visual objects, part 12: ISO base media file format

ITU-T Rec. T.35:2000 SERIES T: Terminals for Telematic Services Procedure for the allocation of ITU-T defined codes for non-standard facilities

T/UWA 005.1-2022 高动态范围（HDR）视频技术 第 1 部分：元数据及适配。

## 3. 术语和定义

下列术语和定义适用于本文件。

### 3.1 HDR Vivid

T/UWA 005.1-2022规定的HDR技术规范，及配套衍生技术的代称。

### 3.2 HDR Vivid 播放设备

能够进行视频解码并依据T/UWA 005.1-2022完成元数据和相应图像处理，最后通过数字视频接口进行输出的设备。

### 3.3 HDR Vivid 显示设备

够依据T/UWA 005.1-2022完成元数据和相应图像处理，并进行图像显示的设备。

## 4. 缩略语

下列缩略语适用于本文件。

HDR 高动态范围（High Dynamic Range）

EOTF 电光转换函数（Electro-Optical Transfer Function）

OETF 光电转换函数（Opto-Electrical Transfer Function）

PQ 基于人眼特性的感知量化方法（Perceptual Quantizer）

HLG 基于混合对数伽马量化方法（Hybrid Log-Gamma）

DASH 基于HTTP的动态自适应流媒体（Dynamic Adaptive Streaming over HTTP）

HLS 基于HTTP的自适应码率流媒体传输（HTTP Live Streaming）

## 5.概述

HDR Vivid视频流包含符合T/UWA 005.1-2022规定的动态元数据流，解码器需要解析该HDR Vivid视频流中的每一帧动态元数据信息，终端需要使用动态元数据信息，根据T/UWA 005.1-2022做相应的处理，如下图所示：



图1 HDR Vivid视频流处理框图

其中HDR Vivid码流生成器负责生成HDR Vivid码流，它的编码模块通过编码器生成ES流，它的封装模块将ES流封装成需要的格式，比如MP4文件，HLS流，和DASH流。

## 6.HDR Vivid 的 ES 流封装

### 6.1.总体要求

HDR Vivid ES流打包时每一帧视频必须包含且仅包含该帧对应的动态元数据。

### 6.2.HEVC/H.265 以及 VVC/H.266 ES 流封装

HDR Vivid 每一帧的动态元数据封装应符合ITU-T Rec. T.35 (02/2000) 及下面的要求：

- 1) 动态元数据封装在user\_data\_registered\_itu\_t\_t35(payloadSize)中；
- 2) user\_data\_registered\_itu\_t\_t35(payloadSize)中的itu\_t\_t35\_country\_code为0x26，itu\_t\_t35\_country\_code表示国家码，0x26表示为中国。；
- 3) itu\_t\_t35\_payload()中的terminal\_provide\_code为0x0004，terminal\_provide\_oriented\_code为0x0005。其中terminal\_provide\_code表示为组织或者机构码，0x0004为UWA世界超高清视频产业联盟，terminal\_provide\_oriented\_code为应用码，由组织或者机构自行定义，0x0005为HDR Vivid的版本1，版本演进方案请见表6.4。

ES 封装的语法见表 1：

表1 ES流封装语法表

伪代码	描述符
user_data_registered_itu_t_t35( payloadSize ) {	
<b>itu_t_t35_country_code</b>	0x26
if( itu_t_t35_country_code != 0xFF){	
i = 1	
}	
else {	
<b>itu_t_t35_country_code_extension_byte</b>	b(8)
i = 2	
}	
do {	
<b>itu_t_t35_payload_byte</b>	b(8)
i++	
} while( i < payloadSize )	
}	

其中**itu\_t\_t35\_payload\_byte**中的数据如下表所示

表2 T35 payload数据结构

itu_t_t35_payload () {	描述符
terminal_provide_code	0x0004
terminal_provide_oriented_code	0x0005 (版本v1.0)
dynamic_metadata ()	
}	

### 6.3.AVS2 以及 AVS3 ES 流封装

HDR Vivid AVS2流和HDR Vivid AVS3流每一帧的ES流中每一帧的HDR Vivid动态元数据封装在extension\_data(i)的hdr\_dynamic\_metadata\_extension ()中, 并且hdr\_dynamic\_metadata\_extension ()中的第一个extension\_id为0x5 (4位),第二个hdr\_dynamic\_metadata\_type为0x5 (4位), 具体见下表:

表3高动态范围图像扩展定义1

高动态范围图像元数据扩展定义	描述符
hdr_dynamic_metadata_extension() {	
<b>extension_id</b>	0x5
<b>hdr_dynamic_metadata_type</b>	0x5
while ( next_bits(24) ! '0000 0000 0000 0000 0000 0001' ) {	
<b>extension_data_byte</b>	u(8)
}	
next_start_code()	
}	

其中表4中的**extension\_data\_byte** ( ) 中的具体语法为:

表4 extension\_data\_byte数据表

动态元数据定义	描述符
<b>extension_data_byte</b> ( ) {	
<b>itu_t_t35_country_code</b>	0x26
<b>terminal_provide_code</b>	0x0004
<b>terminal_provide_oriented_code</b>	0x0005 (版本v1.0)
<b>system_start_code</b>	u(8)
if(system_start_code==0x01){	
num_windows=1	
for( w = 0; w < num_windows; w++ ) {	
<b>minimum_maxrgb_pq[w]</b>	u(12)
marker_bit	f(1)
<b>average_maxrgb_pq[w]</b>	u(12)
marker_bit	f(1)
<b>variance_maxrgb_pq[w]</b>	u(12)
marker_bit	f(1)
<b>maximum_maxrgb_pq[w]</b>	u(12)
marker_bit	f(1)
}	
for(w = 0; w < num_windows; w++ ) {	
<b>tone_mapping_enable_mode_flag[w]</b>	u(1)
if(tone_mapping_enable_mode_flag [w]==1){	
<b>tone_mapping_param_enable_num [w]</b>	u(1)
tone_mapping_param_enable_num [w]++	
for(i=0; i< tone_mapping_param_enable_num [w]; i++){	
<b>targeted_system_display_maximum_luminance_pq[i][w]</b>	u(12)
<b>base_enable_flag[i][w]</b>	u(1)
marker_bit	f(1)
if(base_enable_flag[i][w]){	

表4 extension\_data\_byte数据表 (续)

动态元数据定义	描述符
<b>base_param_m_p[i][w]</b>	u(14)
<b>base_param_m_m[i][w]</b>	u(6)
marker_bit	f(1)
<b>base_param_m_a[i][w]</b>	u(10)
<b>base_param_m_b[i][w]</b>	u(10)
marker_bit	f(1)
<b>base_param_m_n[i][w]</b>	u(6)
<b>base_param_K1[i][w]</b>	u(2)
<b>base_param_K2[i][w]</b>	u(2)
<b>base_param_K3[i][w]</b>	u(4)
<b>base_param_Delta_enable_mode[i][w]</b>	u(3)
marker_bit	f(1)
<b>base_param_enable_Delta[i][w]</b>	u(7)
}	
<b>3Spline_enable_flag[i][w]</b>	u(1)
if(3Spline_enable_flag[i][w]){	
<b>3Spline_enable_num[i][w]</b>	u(1)
3Spline_enable_num[i][w]++;	
for(j = 0; j < 3Spline_enable_num[i][w]; j ++ ) {	
<b>3Spline_TH_enable_mode[j][i][w]</b>	u(2)
if((3Spline_TH_mode[j][i][w]==0)   (3Spline_TH_mode[j][i][w]==2)){	
<b>3Spline_TH_enable_MB [j][i][w]</b>	f(8)
}	
marker_bit	f(1)
<b>3Spline_TH_enable[j][i][w]</b>	f(12)
marker_bit	f(1)
<b>3Spline_TH_enable_Delta1 [j][i][w]</b>	f(10)
<b>3Spline_TH_enable_Delta2 [j][i][w]</b>	f(10)
marker_bit	f(1)
<b>3Spline_enable_Strength[j][i][w]</b>	f(8)
}	
}	
}	
}	
<b>color_saturation_mapping_enable_flag[w]</b>	u(1)
if(color_saturation_mapping_enable_flag[w]) {	
<b>color_saturation_enable_num[w]</b>	u(3)
for(i = 0; i < color_saturation_enable_num [w]; i++ ) {	
<b>color_saturation_enable_gain[i][w]</b>	u(8)

表4 extension\_data\_byte数据表（续）

动态元数据定义	描述符
marker_bit	f(1)
}	
}	
}	
}	
stuffing_bit	'1'
while(!byte_aligned())	
stuffing_bit	'0'
}	

**extension\_data\_byte** () 中的terminal\_provide\_cod和terminal\_provide\_oriented\_code分别对应6.2中itu\_t\_t35\_payload () 结构中描述的 terminal\_provide\_code 及 terminal\_provide\_oriented\_code 。terminal\_provide\_code为0x0004（16位），terminal\_provide\_oriented\_code（16位）为当前版本号。

**extension\_data\_byte** () 中的marker\_bit是为了不应出现从任意字节对齐位置开始的21个以上连续的“0”。**extension\_data\_byte** () 中其他语法元素请参考T/UWA 005.1-2022。

#### 6.4.版本后向兼容

HDR Vivid目前保留了4个版本。每个版本可以生成一套独立的动态元数据，封装在同一个ES流中，也就是一个ES流中可以携带多版本的动态元数据，并且多版本可以是不连续的版本号，例如一个ES流可以传版本1的动态元数据和版本2的动态元数据；或者一个ES流传版本1的动态元数据，版本3的动态元数据和版本4的动态元数据。itu\_t\_t35\_payload () 中的terminal\_provide\_oriented\_code 码字用来标识版本。终端设备建议提取支持的最高版本的动态元数据进行后处理。版本号和标识的映射关系见下表。

表5版本号与标识的映射表

HDR Vivid 版本号	itu_t_t35_payload () 中的terminal_provide_oriented_code码字
1.0	0x0005
2.0	0x0006
3.0	0x0007
4.0	0x0008

备注：如果终端不能识别HDR Vivid某个版本，可忽略该版本；如果全部不能识别，可不做HDR Vivid处理。

#### 7.HDR Vivid MP4 文件封装

HDR Vivid MP4文件在保留原本MP4所有结构的基础上（符合ISO/IEC 14496-12(2015-12-15)），1）在视频轨的Sample Description Box(stsd)内的VisualSampleEntry Box中扩展添加一种新的描述HDR Vivid视频码流格式的Box：CUVV Configuration Box。2）修改VisualSampleEntry Box 中的 compressorname 描述字段为“HDR Vivid video”(可选，应用程序主要通过cuvv box来识别是否是HDR Vivid video)。

新增的CUVV box描述如下：

表6 CUVV box描述表

class CUVVConfigurationBox extends Box('cuvv')	描述符
{	
unsigned int (16) cuva_version_map;	0x0009 (版本号信息)
unsigned int (16) terminal_provide_code;	0x0004
unsigned int (16) terminal_provide_oriented_code;	0x0008 (最高版本号)
const unsigned int (32)[4] reserved = 0;	
}	

新增box的 four-character-code为cuvv。内部结构描述了HDR Vivid视频的版本号信息：16位的 cuva\_version\_map，每一位表示一个版本号，一共可以表示16个版本号，其中高位表示高版本号，低位表示低版本号，例如cuva\_version\_map为0x0009该码流有版本4码流和版本1码流（9为1001），另外两位描述符分别对应6.2中itu\_t\_t35\_payload ()结构中描述的 terminal\_provide\_code 及 terminal\_provide\_oriented\_code。terminal\_provide\_code为0x0004（16位），terminal\_provide\_oriented\_code（16位）为当前码流中包含的最高版本对应的值，例如当cuva\_version\_map为0x0009时，当前码流包含的最高版本为版本4，根据6.4节中表6.4可知版本4对应的terminal\_provide\_oriented\_code应为0x0008。

在MP4文件中添加cuvv的具体方法为在moov->trak(video)->mdia->minf->stbl->stsd内的VisualSampleEntry Box中，添加一个cuvv Box。

参考ISO/IEC 14496-12(2015-12-15)，VisualSampleEntry描述如下：

表7 VisualSampleEntry描述表

class VisualSampleEntry(codingname) extends SampleEntry (codingname){	描述符
unsigned int(16) pre_defined = 0;	
const unsigned int(16) reserved = 0;	
unsigned int(32)[3] pre_defined = 0;	
unsigned int(16) width;	
unsigned int(16) height;	
template unsigned int(32) horizresolution = 0x00480000; // 72 dpi	
template unsigned int(32) vertresolution = 0x00480000; // 72 dpi	
const unsigned int(32) reserved = 0;	
template unsigned int(16) frame_count = 1;	
string[32] compressorname;	
template unsigned int(16) depth = 0x0018;	
int(16) pre_defined = -1;	
// other boxes from derived specifications	
CleanApertureBox clap; // optional	
PixelAspectRatioBox pasp; // optional	

}

其中新增的cuvv box放到“other boxes from derived specifications”位置，在其它扩展box之后添加cuvv box。

例：在H.265的MP4文件中，VisualSampleEntry为'hvc1'或'hev1'，已经存在hvcC box用于描述视频编码码流信息，HDR Vivid MP4文件则是在hvcC box后面添加一个新的box：cuvv Box；其余结构除了compressorname 描述修改为“CUVA HDR Video”之外，全部保持不变。

注：HDR Vivid MP4文件示例：

The screenshot displays the internal structure of an MP4 file named 'HDR\_vivid\_versionmap1.mp4'. On the left, a tree view shows the file's hierarchy, with 'hvc1' and 'cuvv' boxes highlighted in red. The central pane shows the 'HEVC Sample Entry' details, including 'Detailed-Information' with fields like Width, Height, and Compression name 'CUVA HDR Video'. The right pane shows the raw hex data for the 'hvc1' and 'cuvv' boxes.

图2 HDR Vivid MP4文件描述图

## 8.HDR Vivid HLS 流封装

### 8.1.总体要求

HDR Vivid HLS流的播放列表文件（playlist file）除了增加m3u8文件关于HDR Vivid视频信息外，其他均符合HLS流标准 HTTP Live Streaming 2nd Edition的规定。

### 8.2.HLS 流封装

HLS流中使用#EXT-X-STREAM-INF或者EXT-X-I-STREAM-INF标识视频或者音频的属性，HDR Vivid HLS流的封装在保持原m3u8描述的基础上，对于每一路视频，新增一条#EXT-X-STREAM-INF或者EXT-X-I-STREAM-INF描述信息，其中CODECS标签内原来描述视频信息的部分更改为HDR Vivid描述符，其中HDR Vivid描述符的格式定义为：[CUVAHDR\_video].[CUVA\_Version\_map]，其中CUVAHDR\_video标识表示的是HDR Vivid码流，HLS流为mp4或者TS封装时，CUVAHDR\_video值都为cuvv。CUVA\_Version\_map表示HDR Vivid的版本号信息，CUVA\_Version\_map中的每一位表示一个版本号，其中高位表示高版本号，低位表示低版本号，例如CUVA\_Version\_map为1101表示该码流有版本4码流，版本3码流和版本1码流。其它的音视频属性信息还是直接使用原来的描述。其中VIDEO-RANGE描述项符合《HTTP Live Streaming 2nd Edition draft-pantos-hls-rfc8216bis-02》描述。

HLS M3u8 描述示例：

/\*新增的两条不同码率音视频属性描述\*/

```
#EXT-X-STREAM-INF:BANDWIDTH=5120000,VIDEO-RANGE=PQ,CODECS="cuvv.1101,mp4a.40.29",FRAME-RATE=50.000,RESOLUTION=3840x2160,AUDIO="aac"
```

```

low/video.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=7680000,VIDEO-RANGE=PQ,CODECS="cuvv.1101,mp4a.40.29",FRAME-RATE=50.000,RESOLUTION=3840x2160,AUDIO="aac"
high/video.m3u8
/*原始的两条不同码率音视频属性描述*/
#EXT-X-STREAM-INF:BANDWIDTH=5120000,VIDEO-RANGE=PQ, CODECS="hvc1.2.4.L153.b0,mp4a.40.29",FRAME-RATE=50.000,RESOLUTION=3840x2160,AUDIO="aac"
low/video.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=7680000,VIDEO-RANGE=PQ,CODECS="hvc1.2.4.L153.b0,mp4a.40.29",FRAME-RATE=50.000,RESOLUTION=3840x2160,AUDIO="aac"
high/video.m3u8

```

## 9.HDR Vivid DASH 封装

### 9.1.总体要求

HDR Vivid DASH (Dynamic Adaptive Streaming over HTTP) 流的mpd文件除了增加Representation关于HDR Vivid视频信息外, 其他均符合MPEG-DASH国际标准ISO/IEC FDIS 23009要求。

### 9.2.DASH 流封装

DASH流的mpd文件的Representation字段标识关于视频或者音频的属性, 包含了帧率, 分辨率, 码率, 编码器, 带宽等信息。

HDR Vivid DASH流的封装在保持原Representation描述的基础上, 对于每一路视频, 新增一条Representation描述信息, 其中codecs标签内原来描述视频信息的部分更改为HDR Vivid描述符, 其中HDR Vivid描述符的格式定义为: [CUVAHDR\_video].[CUVA\_Version], 其中HDR Vivid\_video标识表示的是HDR Vivid码流, 当DASH流为mp4或者TS封装时, HDR VividR\_video都为cuvv。CUVA\_Version\_map表示HDR Vivid的版本号信息, CUVA\_Version\_map中的每一位表示一个版本号, 其中高位表示高版本号, 低位表示低版本号, 例如CUVA\_Version\_map为1110表示该码流有版本4码流, 版本3码流以及版本1码流。DASH 音视频描述分离, 所以codecs内只需描述视频信息, 新增的Representation也仅在每个视频轨上新增一条即可。

HDR Vivid DASH码流mpd文件描述示例如下:

```

<Representation id="bbb_30fps_1024x576_2500k" codecs="cuvv.110" bandwidth="3134488"/> //新增
<Representation id="bbb_30fps_1280x720_4000k" codecs="cuvv.110" bandwidth="4952892" /> //新增
<Representation id="bbb_30fps_1024x576_2500k" codecs=" hvc1.2.4.L153.b0" bandwidth="3134488" />
<Representation id="bbb_30fps_1280x720_4000k" codecs=" hvc1.2.4.L153.b0" bandwidth="4952892" />

```

## 10.HDR Vivid DVB TS 流封装要求

### 10.1.总体要求

HDR Vivid DVB TS流除了增加component\_descriptor中关于HDR Vivid码流的描述, 其他语法均符合ETSI EN 300 468 V1.16.1 (2019-08)要求, 具体修改的表为PMT表。

当节目含有HDR服务, 而元数据缺失, 应进行容错处理。

### 10.2.TS 流的封装

在保持component\_descriptor原始描述的语法基础上，新增加以下字段来指定使用的HDR Vivid的动态元数据。除了新增的两个描述符信息，其它结构不变。

1) 在PMT表的视频描述信息中，新增一个注册描述符registration\_descriptor()，内部字段定义如下：

表8 registration\_descriptor注册描述符表

registration_descriptor ()	描述符
{	
unsigned int (8) descriptor_tag ;	0x05
unsigned int (8) descriptor_length;	0x04
unsigned int (32) Format_identifier ;	'cuvv'
}	

descriptor\_tag 为 0x05，为注册描述符 TAG。

descriptor\_length 为 0x04，为'cuvv'描述符长度。

Format\_identifier 值必须为 0x63757676('cuvv')值,这个值可以用于快速甄别 TS 码流是否为 HDR Vivid 码流。

2) 在 PMT 表的视频描述信息中，新增一个用户自定义描述符，可以用于描述 HDR Vivid 版本信息等，定义如下：

表9用户自定义描述符表

CUVV_video_stream_descriptor()	描述符
{	
unsigned int (8) descriptor_tag;	0xF3
unsigned int (8) descriptor_length;	0x0A
unsigned int (32) cuvv_tag ;	'cuvv'
unsigned int (16) cuva_version_map;	0x0005 (版本号信息)
unsigned int (16) terminal_provide_code;	0x0004
unsigned int (16) terminal_provide_oriented_code;	0x0007 (最高版本号)
}	

descriptor\_tag 使用用户自定义 0xF3。

descriptor\_length 为 10。

cuvv\_tag 值为 0x63757676('cuvv')，主要用户 0xF3 用户自定义 tag 冲突时，进一步标识该描述符内容为 CUVV 描述信息。

CUVA\_Version\_map 表示 HDR Vivid 的版本号信息，总共 16 位，CUVA\_Version\_map 中的每一位表示一个版本号，其中高位表示高版本号，低位表示低版本号，例如 CUVA\_Version\_map 为 0x0005(5 对应 0101)表示该码流有版本 3 码流和版本 1 码流。

最后两位描述符分别对应 6.2 中 `itu_t_t35_payload()` 结构中描述的 `terminal_provide_code` 及 `terminal_provide_oriented_code`。`terminal_provide_code` 为 0x0004 (16 位), `terminal_provide_oriented_code` (16 位) 为当前码流中包含的最高版本对应的值, 例如当 `cuva_version_map` 为 0x0005 时, 当前码流包含的最高版本为版本 3, 根据 6.4 节中表 5 可知版本 3 对应的 `terminal_provide_oriented_code` 应为 0x0007。

### 10.3.TS 流的丢包处理

HDR Vivid DVB TS流出现无法同步, 或者丢包3个以上, 需要重新同步。

## 11.终端应用说明

### 11.1.终端系统组成

#### 11.1.1.功能模块

HDR Vivid终端部分包含视频接收解码和视频内容呈现两部分功能, 其框图如下图所示。

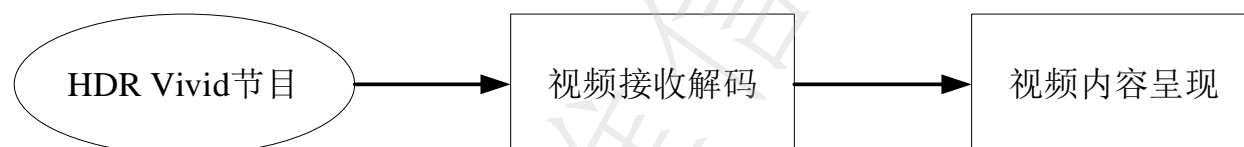


图3 HDR Vivid终端功能框图

视频接收解码通过播放设备实现, 一般主要指带有网口、同轴接口、USB等输入接口并带有视频解码能力的设备, 包括机顶盒, 播放器, 或带解码能力的显示设备等。HDR Vivid播放设备指能够接收、解码、输出HDR Vivid视频内容和动态元数据的视频播放设备。

视频内容呈现通过显示设备实现, 一般主要指电视, 投影仪, 显示器, 笔记本电脑, 手机等具备图像呈现功能的设备。根据其对于HDR Vivid的支持能力, 显示设备可以分为以下几种类别:

表10显示设备分类

设备类型	HDR Vivid显示设备	HDR显示设备	SDR显示设备
说明	符合HDR Vivid规范, 能够接收、处理HDR Vivid元数据并显示视频信号。	仅能接收、解码、显示BT.2100定义的PQ-HDR或HLG-HDR视频信号及静态元数据。	不支持BT.2100定义的HDR信号, 仅能接收、解码、显示BT.2020或BT.709或BT.601定义的SDR视频信号。

#### 11.1.2.应用方式

终端系统可以有以下两种应用方式:

##### a) 一体机应用方式

显示设备直接从视频信号分发传输网络接收HDR Vivid视频信号, 在显示设备内部完成解码、播放、显示呈现全过程, 就是一体机应用方式。

##### b) 播放器应用方式

显示设备通过播放设备间接从视频节目分发传输网络接收HDR Vivid视频信号, 在播放设备内进行HDR Vivid视频信号的解码播放, 显示设备进行显示呈现。播放设备与显示设备通过数字视频接口联接及传输HDR Vivid视频信号。该应用方式为播放器应用方式

播放设备仅需要支持播放器应用方式。显示设备应支持一体机方式或播放器方式。

## 11.2.HDR Vivid 信号终端解码与呈现

## 11.2.1.基本要求

播放设备需要至少提供一种适配模式,在该模式下仅进行基于本规范要求的处理和其他必要的图像处理,以确保能够输出的图像和信息满足本标准的要求。

显示设备需要至少提供一种适配模式,能够按照本标准定义的认证模式之要求进行图像处理,以确保HDR Vivid视频图像能够按照本标准要求进行呈现。

## 11.2.2.播放器应用方式的适配模式

当HDR Vivid的解码和显示由播放设备与显示设备共同完成时,二者有两种适配模式:

## a) 接收端适配模式

系指播放设备将HDR Vivid信号及动态元数据送给显示设备,由显示设备按照T/UWA 005.1-2022要求进行色彩动态范围转换过程和色彩校正过程处理的适配模式。接收端适配模式下,当显示设备接收到的视频图像出现不同传输特性曲线、不同色域或色彩空间转换矩阵等信息变化时,画面应当保持稳定不出现视觉可见的闪烁、黑屏等现象。

## b) 监视器适配模式

系指播放设备根据显示设备提供的动态范围信息(EDID)按照T/UWA 005.1-2022要求完成色彩动态范围转换过程和色彩校正过程处理,并将处理后的图像传送给显示设备的适配模式。这种模式下,显示设备接收到播放设备发送的图像信号后,不再进行色彩动态范围转换过程和色彩校正过程处理,只将该图像信号进行传输特性曲线转换和其他必要处理后进行显示。

HDR Vivid播放设备需要完全支持上述两种适配模式;当HDR Vivid显示设备具备数字视频输入接口且该接口支持HDR Vivid格式时,需要支持至少上述两种适配模式之其中一种。HDR Vivid播放设备应该优先以接收端适配模式连接HDR Vivid显示设备;当HDR Vivid显示设备仅支持监视器适配模式时HDR Vivid播放设备才以监视器适配模式连接HDR Vivid显示设备。

## 11.2.3.PQ HDR 信号的互联适配

当HDR Vivid播放设备接收到类型为PQ HDR的HDR Vivid信号时,需要根据对接的显示设备类型进行信号格式的转换。具体方式如下表所示:

表 11显示设备的PQ信号格式转换表

显示设备类型	HDR Vivid显示设备		HDR显示设备	SDR显示设备
	监视器适配模式	接收端适配模式		
播放设备输出信号	经处理的PQ HDR信号和HDR Vivid VS-IF信息帧	PQ HDR信号和HDR Vivid动态元数据VS-EMDS信息帧	1) PQ HDR信号及静态元数据(如果视频信号中包含静态元数据) 2) 或经处理的PQ HDR信号	经处理转换为SDR信号(输出色域优先为BT2020其次BT709最次BT601)

## 11.2.4.HLG HDR 信号的互联适配

当HDR Vivid播放设备接收到类型为HLG HDR的HDR Vivid信号时,需要根据对接的显示设备类型进行信号格式的转换。具体方式如下表所示:

表 12显示设备的HLG信号格式转换表

显示设备类型	HDR Vivid显示设备	HDR显示设备	SDR显示设备

	监视器适配模式	接收端适配模式		
播放设备输出信号	经处理的PQ HDR信号和HDR Vivid VS-IF信息帧	HLG HDR信号和HDR Vivid动态元数据VS-EMDS信息帧（ <sup>1</sup> 如果视频信号中存在HDR Vivid动态元数据）	1) HLG HDR（ <sup>2</sup> PQ HDR）信号及静态元数据（如果视频信号中包含静态元数据） 2) 或经处理的PQ HDR信号	经处理转换为SDR信号（输出色域优先为BT2020其次BT709其次BT601）

<sup>1</sup>HDR Vivid对于HLG HDR信号是否需携带动态元数据不做强制要求。

<sup>2</sup>部分HDR显示设备不能支持HLG HDR格式，只能支持PQ HDR格式，这种情况下，播放设备应将HLG HDR信号转为PQ HDR信号送给HDR显示设备显示。

### 11.3.动态元数据 HDMI 传输要求

#### 概述

HDR Vivid播放设备与HDR Vivid显示设备通过HDMI进行连接和图像数据传输时，应当按照本章节要求进行动态元数据传输。

基于HDMI传输的HDR Vivid信号，信号的电-光传输特性由VS-IF或VS-EMDS中transfer\_character字段确定，信号的色域为BT.2020。

当HDR Vivid播放设备与HDR Vivid显示设备建立连接后，播放设备通过查询显示设备的EDID中符合本规范11.3.2规定的VS-VDB数据块来了解显示设备对HDR Vivid标准的支持能力。

当显示设备的VS-VDB表明能够支持HDR Vivid接收端适配模式时，播放设备应将HDR Vivid的动态元数据封装在符合本规范11.3.4规定的VS-EMDS信息帧中，配合HDR Vivid视频图像发送给显示设备。

当显示设备的VS-VDB表明能够支持HDR Vivid监视器适配模式时，播放设备应根据显示设备VS-VDB中提供的最大显示亮度和最小显示亮度信息，将HDR Vivid的视频内容根据T/UWA 005.1-2022第9章及附录B.3的规定进行适配处理后发送给显示设备，并同时向显示设备发送本规范11.3.3规定的VS-IF信息。

#### 11.3.1.VS-VDB 信息定义

HDR Vivid显示设备在HDMI EDID中采用VS-VDB数据块来标识是否具备HDR Vivid动态元数据接收能力。

##### 11.3.1.1.语法

HDR Vivid VS-VDB数据块信息语法如下表所示：

表 13 VS-VDB数据块信息语法表

Byte\Bit	7	6	5	4	3	2	1	0
0	Tag Code= (0x07)			Length (=27)				
1	Extended Tag Code=0x01 (VSVDB)							
2	IEEE OUI/CID (0x03)							
3	IEEE OUI/CID (0x75)							
4	IEEE OUI/CID (0x04)							
5	system_start_code[7...0]							
6	version_code[3...0]			0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)
7	display_maximum_luminance [31...0]							
8								
9								
10								
11	display_minimum_luminance [15...0]							
12								
13	monitor_mode_support	rx_mode_support	0 (Reservd)					
14~27	0 (Reservd)							

### 11.3.1.2.语义

#### 系统起始码 system\_start\_code

8位无符号整数。表示终端支持的HDR Vivid版本号,当前版本赋值为1。

如果当前版本号高于码流中接收到的动态元数据所包含的system\_start\_code, 则按照HDR Vivid VS-EMDS包数据结构将动态元数据进行打包; 否则在当前发送端设备上根据HDR Vivid元数据标准进行显示适配处理, 然后将进行过适配处理之后的数据发送到接收端。

#### 系统版本码 version\_code

4位无符号整数。表示终端的HDR Vivid系统版本号, 当前版本赋值为1。发送端按照接收端的版本发送相应格式的动态元数据信息。

#### 设备最高显示亮度 display\_maximum\_luminance

32位无符号整数, 表示显示设备在D65色温下可以显示的最大亮度, 单位是0.0001尼特。当码字为0x00000001时, 表示0.0001尼特。MaxDisplay最终取值为display\_maximum\_luminance /10000。当显示设备支持监视器适配模式时, 显示设备需要为display\_maximum\_luminance设置适当数值, 以便播放设备使用该信息进行图像处理; 当显示设备不支持监视器适配模式时, 应该将display\_maximum\_luminance设置为0。

#### 设备最小显示亮度 display\_minimum\_luminance

16位无符号整数, 表示显示设备在D65色温下可以显示的最小亮度, 单位是0.0001尼特。当码字为0x0001时, 表示0.0001尼特。MinDisplay最终取值为display\_minimum\_luminance /10000。当显示设备支持监视器适配模式时, 显示设备需要为display\_minimum\_luminance设置适当数值, 以便播放设备使用该信息进行图像处理; 当显示设备不支持监视器适配模式时, 应该将display\_minimum\_luminance设置为0。

**监视器适配模式支持标识monitor\_mode\_support**

1位无符号整数。表示显示终端对监视器适配模式的支持能力，monitor\_mode\_support=1表示显示终端支持监视器适配模式，monitor\_mode\_support=0表示显示终端不支持监视器适配模式。

**接收端适配模式支持标识rx\_mode\_support**

1位无符号整数。表示显示终端对接收端适配模式的支持能力，rx\_mode\_support=1表示显示终端支持接收端适配模式，rx\_mode\_support=0表示显示终端不支持接收端适配模式。

**11.3.2.VS-IF 信息定义****11.3.2.1.语法**

在HDR Vivid 监视器适配模式下，采用HDMI中VS-IF信息帧从播放设备向显示设备传输静态元数据。VS-IF信息帧语法如下表所示：

表14 VS-IF信息帧语法表

Byte\Bit	7	6	5	4	3	2	1	0
HB0	VSIF Type Code =0x81							
HB1	VSIF Version=0x01							
HB2	0	0	0	Payload Length=27				
PB00	checksum							
PB01	IEEE OUI/CID (0x03)							
PB02	IEEE OUI/CID (0x75)							
PB03	IEEE OUI/CID (0x04)							
PB04	system_start_code[7...0]							
PB05	version_code[3...0]				Monitor_m ode_enable =0x01	transfer_ character=0	0 (Rsvd)	0 (Rsvd)
PB06~PB27	0 (Reserved)							

**11.3.2.2.语义****监视器适配模式支持标识monitor\_mode\_enable**

1位无符号整数。本版本规范要求monitor\_mode\_enable固定为1，表示此时的HDR Vivid信号的互联适配处理方式为监视器适配模式。

**图像传输特性 transfer\_character**

1位无符号整数。表示图像的传输特性，本版本规范要求transfer\_character固定为0，表示监视器适配模式下，图像传输特性为ST2084 EOTF曲线。

**11.3.3.VS-EMDS 信息定义**

在HDR Vivid 接收端适配模式下，采用HDMI中VS-EMDS信息帧从播放设备向显示设备传输动态元数据。

**11.3.3.1.语法**

HDR Vivid的VS-EMDS信息语法如下表所示：

表15 VS-EMDS信息语法表

Byte\ Bit	7	6	5	4	3	2	1	0
HB0	0	1	1	1	1	1	1	1
HB1	1 (First)	0 (Last)	(Rsvd )	(Rsvd )	(Rsvd )	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)
HB2	0 (Sequence_Index)							
PB0	1 (New)	0 (End)	1 (DS_Type)	0 (AFR )	1 (VFR)	1 (Sync)	0 (Rsvd)	
PB01	0 (Rsvd)							
PB02	0 (Organization_ID)							
PB03	0x00 (Data_Set_Tag_MSB)							
PB04	0x02 (Data_Set_Tag_LSB)							
PB05	0x00 (Data_Set_Length_MSB)							
PB06	0x38 (Data_Set_Length_LSB)							
PB07	IEEE OUI/CID (03)							
PB08	IEEE OUI/CID (75)							
PB09	IEEE OUI/CID (04)							
PB10	system_start_code[7...0]							
PB11	version_code[3...0]				minimum_maxrgb_pq [11...8]			
PB12	minimum_maxrgb_pq [7...0]							
PB13	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	average_maxrgb_pq [11...8]			
PB14	average_maxrgb_pq [7...0]							
PB15	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	variance_maxrgb_pq [11...8]			
PB16	variance_maxrgb_pq [7...0]							
PB17	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	maximum_maxrgb_pq [11...8]			
PB18	maximum_maxrgb_pq [7...0]							
PB19	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	targeted_system_display_maximum_luminance_pq[11...8]			
PB20	targeted_system_display_maximum_luminance_pq[7...0]							
PB21	Transfer_Character	base_enable flag[0]	base_param_m_p[13...8]					
PB22	base_param_m_p[7...0]							

Byte\ Bit	7	6	5	4	3	2	1	0
PB23	0 (Rsvd)	0 (Rsvd )	base_param_m_m[5...0]					
PB24	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd)	base_param_m_a[9,8]	
PB25	base_param_m_a[7...0]							
PB26	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd)	base_param_m_b[9...8]	
PB27	base_param_m_b[7...0]							
HB0	0	1	1	1	1	1	1	1
HB1	0 (First)	0(Last)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)
HB2	1 (Sequence_Index)							
PB0	0 (Rsvd)	0 (Rsvd )	base_param_m_n[5...0]					
PB1	0 (Rsvd)	0 (Rsvd )	base_param_K1[1..0]		base_param_K2[1..0]		base_param_K3[1..0]	
PB2	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	base_param_Delta_enable_mode[2..0]		
PB3	0 (Rsvd)	base_param_enable_Delta[6...0]						
PB4	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	3Spline_enable_nu m		3Spline _enable_flag[0 ]	3Spline_TH_enable_mode0[1, 0]	
PB5	3Spline_TH_enable_MB0[7...0]							
PB6	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	3Spline_TH_enable0[11...8]			
PB7	3Spline_TH_enable0[7...0]							
PB8	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd)	3Spline_TH_enable_0Delta1[ 9,8]	
PB9	3Spline_TH_enable_0Delta1[7...0]							
PB10	0 (Rsvd)	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd )	0 (Rsvd)	3Spline_TH_enable_0Delta2[ 9,8]	

Byte\ Bit	7	6	5	4	3	2	1	0
		)	)	)	)			
PB11	3Spline_TH_enable_0Delta2[7...0]							
PB12	3Spline_enable_Strength0[7...0]							
PB13	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	3Spline_TH_enable_mode1[1,0]	
PB14	3Spline_TH_enable_MB1[7...0]							
PB15	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	3Spline_TH_enable1[11...8]			
PB16	3Spline_TH_enable1[7...0]							
PB17	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	3Spline_TH_enable_1Delta1[9,8]	
PB18	3Spline_TH_enable_1Delta1[7...0]							
PB19	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	3Spline_TH_enable_1Delta2[9,8]	
PB20	3Spline_TH_enable_1Delta2[7...0]							
PB21	3Spline_enable_Strength1[7...0]							
PB22	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	color_saturation_enable_num[2...0]		
PB23	color_saturation_enable_gain0[7...0]							
PB24	color_saturation_enable_gain1[7...0]							
PB25	color_saturation_enable_gain2[7...0]							
PB26	color_saturation_enable_gain3[7...0]							
PB27	color_saturation_enable_gain4[7...0]							
HB0	0	1	1	1	1	1	1	1
HB1	0 (First)	1 (Last)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)	0 (Rsvd)
HB2	2 (Sequence_Index)							
PB0	color_saturation_enable_gain5[7...0]							
PB1	color_saturation_enable_gain6[7...0]							
PB2	color_saturation_enable_gain7[7...0]							
PB3	graphic_source_display_value[7...0]							
PB4	0 (Reservd)							
PB5	max_display_mastering_luminance,MSB							

Byte\ Bit	7	6	5	4	3	2	1	0
PB6	max_display_mastering_luminance,LSB							
PB7~P B27	0 (Reservd)							

### 11.3.3.2. 语义

#### 系统版本码 **version\_code**

4位无符号整数。表示系统版本号,当前版本赋值为1。

#### 图像传输特性 **transfer\_character**

1位无符号整数。表示图像的传输特性, transfer\_character为0, 表示传输特性为ST2084 EOTF曲线; transfer\_character为1, 表示传输特性为Hybrid Log-Gamma (HLG) OETF曲线。

#### 图像固定曲线区间参数 **graphic\_source\_display\_value\_pq**

8位无符号整数, 表示播放设备传输给显示设备的图形对应的PQ域非线性归一化内容显示亮度码值, 图形内容的色域及传输特性与视频一致, 传输特性由图像传输特性transfer\_character确定。graphic\_source\_display\_value\_pq为0时, 代表没有图形传输。图像固定曲线源亮度参数graphic\_source\_value\_pq取值为graphic\_source\_display\_value\_pq/255。使用参考11.3.4接收端适配模式显示端适配过程的图形处理。

#### 图像主监视器最大亮度 **max\_display\_mastering\_luminance**

16位无符号整数, 合并max\_display\_mastering\_luminance,LSB和max\_display\_mastering\_luminance,MSB获得。其中0x0001表示1nit, 0xFFFF表示65535nits。其余参数参考T/UWA 005.1-2022中动态元数据相关语法。

### 11.3.4. 接收端适配模式显示端适配过程的图形处理

- 接收端接收到 VS-EMDS包之后, 根据HDR Vivid VS-EMDS包数据结构, 从中提取出HDR Vivid元数据, 以及图像固定曲线区间参数;
- 参照 T/UWA 005.1-2022的第9章内容生成中间色调映射曲线P, 包含基础曲线参数 $P_{tone\_mapping}$ , 包含m\_p、m\_m、m\_n、m\_a、m\_b、K1、K2、K3; 一次样条曲线参数 $P_{1spline}$ , 包含MB[0][0]和TH3[0]; 三次样条曲线参数 $P_{3spline}$ , 包含TH1[3Spline\_num]、TH2[3Spline\_num]、TH3[3Spline\_num]、MA[2][3Spline\_num]、MB[2][3Spline\_num]、MC[2][3Spline\_num]、MD[2][3Spline\_num];
- 若graphic\_source\_value\_pq=0, 则将最终色调映射曲线S设置为P;
- 否则, 若graphic\_source\_value\_pq>0, 则图形显示的目标亮度为

$$graphic\_target\_value\_pq = \max(\min(graphic\_source\_value\_pq, MaxDisplayPQ \times 0.9), 0.5081)$$

将S中的一次样条曲线参数 $S_{1spline}$ , 包含MB[0][0]和TH3[0]设置为:

$$MB[0][0] = graphic\_target\_value\_pq / graphic\_source\_value\_pq;$$

$$TH3[0] = graphic\_source\_value\_pq;$$

$$base\_offset=0$$

将S中的基础曲线参数 $S_{tone\_mapping}$ , 包含s\_m\_p、s\_m\_m、s\_m\_n、s\_m\_a、s\_m\_b、s\_K1、s\_K2、s\_K3设置为基础曲线参数 $P_{tone\_mapping}$ , 包含m\_p、m\_m、m\_n、m\_a、m\_b、K1、K2、K3;

参照T/UWA 005.1-2022中9.3节, 按照如下步骤生成三次样条曲线参数 $S_{3spline}$ ,

步骤1:设置 $TH1[1] = TH3[0]$ , 并计算 $TH2[1] = TH1[1] + B$ ,  $TH3[1] = TH2[1] + C$ , 其中B默认值为0.15, C为B/2;若 $TH3[1] > MaxSource$ , 则 $TH3[1] = MaxSource$ ;  $TH2[1] = TH1[1] + 2 * (TH3[1] - TH1[1]) / 3$ ; 然后通过T/UWA 005.1-2022中9.3.3.1或9.3.3.2生成三次样条曲线参数S3spline, 并判断该三次样条曲线是否单调递增(见11.3.4.1), 若是, 则执行步骤2;若否, 则执行步骤3;

步骤2:当 $TH1[3Spline\_num]$ 大于 $TH3[1]$ 时, 将该三次样条曲线参数P3spline, 包含 $TH1[3Spline\_num]$ 、 $TH2[3Spline\_num]$ 、 $TH3[3Spline\_num]$ 、 $MA[2][3Spline\_num]$ 、 $MB[2][3Spline\_num]$ 、 $MC[2][3Spline\_num]$ 、 $MD[2][3Spline\_num]$ 赋值给三次样条曲线参数S3spline; 否则,  $TH1[2]$ 、 $TH2[2]$ 、 $TH3[2]$ 、以及第2条三次样条参数均为0, 并将metadata中3spline\_num设为1, 结束三次样条曲线参数生成过程。

步骤3:设置 $TH3[1] = MaxSource$ ;  $TH2[1] = TH1[1] + 2 * (TH3[1] - TH1[1]) / 3$ , 生成三次样条曲线参数S3spline, 并判断此时三次样条曲线是否单调递增(见11.3.4.1): 若是, 则执行步骤4; 若否, 则执行步骤5。

步骤4: 用二分查找法在 $[TH1[1] + B + C, MaxSource]$ 中寻找一个最小的TH3, 通过T/UWA 005.1-2022中9.3.3.1或9.3.3.2生成三次样条曲线参数S3spline, 使曲线单调递增(见11.3.4.1)。最多查找10次, 结束三次样条曲线参数生成过程。

步骤5:  $TH3[1] = MaxSource$ ,  $TH2[1] = TH3[1]$ ,

$MA[0][1] = graphic\_target\_value\_pq$ ,  $MB[0][1] = (VA2 - MA[0][1]) / (TH2[1] - TH1[1])$ ,  $MC[0][1]$ ,  $MD[0][1]$ ,  $MA[1][1]$ ,  $MB[1][1]$ ,  $MC[1][1]$ ,  $MD[1][1]$ 均为0, 并结束三次样条曲线参数生成过程。

#### 11.3.4.1. 判断三次样条曲线是否单调递增的过程

一个三次样条区间包含两个三次样条曲线, 具体方程如下:

$$F(L) = MD[0][n] \times (L - TH1[n])^3 + MC[0][n] \times (L - TH1[n])^2 + MB[0][n] \times (L - TH1[n]) + MA[0][n]$$

其中L为区间 $[TH1[n], TH2[n]]$ 中的自变量,

$$F(L) = MD[1][n] \times (L - TH2[n])^3 + MC[1][n] \times (L - TH2[n])^2 + MB[1][n] \times (L - TH2[n]) + MA[1][n]$$

其中L为区间 $[TH2[n], TH3[n]]$ 中的自变量,  $0 < n \leq 3Spline\_num$ 。

需要同时满足两个三次样条曲线都单调递增, 具体判断条件如下,

a)第一个三次样条曲线单调递增时要满足的条件为:

若 $TH1[n] \leq (TH1[n] - \frac{MC[0][n]}{3 \times MD[0][n]}) \leq TH2[n]$ , 则需满足:

$$MB[0][n] - \frac{MC[0][n]^2}{3 \times MD[0][n]} \geq 0$$

若 $TH1[n] > (TH1[n] - \frac{MC[0][n]}{3 \times MD[0][n]})$ 或 $(TH1[n] - \frac{MC[0][n]}{3 \times MD[0][n]}) > TH2[n]$ , 则需满足:

$$\begin{cases} MB[0][n] \geq 0 \\ 3 \times MD[0][n] \times (DTH2)^2 + 2 \times MC[0][n] \times (DTH2) + MB[0][n] \geq 0 \end{cases}$$

其中： $DTH2 = (TH2[n] - TH1[n])$

b)第二个三次样条曲线单调递增时要满足的条件为：

若 $TH2[n] \leq (TH2[n] - \frac{MC[1][n]}{3 \times MD[1][n]}) \leq TH3[n]$ ，则需满足：

$$MB[1][n] - \frac{MC[1][n]^2}{3 \times MD[1][n]} \geq 0$$

若 $TH2[n] > (TH2[n] - \frac{MC[1][n]}{3 \times MD[1][n]})$ 或 $(TH2[n] - \frac{MC[1][n]}{3 \times MD[1][n]}) > TH3[n]$ ，则需满足：

$$\begin{cases} MB[1][n] \geq 0 \\ 3 \times MD[1][n] \times (DTH3)^2 + 2 \times MC[1][n] \times (DTH3) + MB[1][n] \geq 0 \end{cases}$$

其中： $DTH3 = (TH3[n] - TH2[n])$ 。

## 12.后期制作的应用说明

### 12.1.输入输出建议

后期制作主要完成的工作是动态元数据的提取、调节和文件的封装。

后期制作输入的基本要求是BT.2020色域。可以是线性域的RGB信号，PQ域的RGB信号，或者PQ域的YUV信号。线性域建议位宽至少16bit，PQ域至少12bit。YUV的采样格式建议YUV444或者YUV422。

输出要求是PQ域的YUV420、YUV422或者YUV444的10bit位宽的视频数据加上动态元数据。若输出目的为直接分发，建议输出指定编码格式的码流文件，将动态元数据嵌在的SEI报头中。若输出目的为归档，建议输出MXF文件。视频内容采用高码流格式单帧存放，如XAVC。动态元数据嵌在XAVC的每帧SEI数据中。

### 12.2.画面质量监控

后期制作需要支持两路显示输出。第一路，视频源需要输出到专业的监视器上显示，作为高质量的参考。第二路，采用HDR Vivid动态元数据对视频进行后处理（色调映射），并将后处理的结果输出到消费级终端设备显示，如一台500 nit的HDR Vivid电视，作为终端显示器的参考。

### 12.3.动态元数据生成模式

动态元数据的生成包括两种模式：自动模式和导演模式。

#### 12.3.1.自动模式

自动模式对视频的每一帧提取HDR Vivid动态元数据，进行HDR Vivid实时后处理，将结果输出到显示设备播放。自动模式需要集成T/UWA 005.1-2022规定的动态元数据自动提取算法和后处理算法。

自动模式的界面要支持视频源的读入和播放、动态元数据的提取、HDR Vivid后处理以及后处理结果的输出显示。

## 12.3.2.导演模式

## 12.3.2.1.概述

导演模式的目的是帮助调色师手动调节每一个场景或者每一帧的动态元数据,达到调色师最满意的效果。基本要求是提供HDR Vivid色调映射曲线的调节控件,并且在调色师调节的过程中,能够显示当前的色调映射曲线,同时可以基于当前的动态元数据或者曲线进行实时的后处理并输出显示。

导演模式需要集成自动模式的动态元数据提取算法、后处理算法和导演模式曲线调节算法。

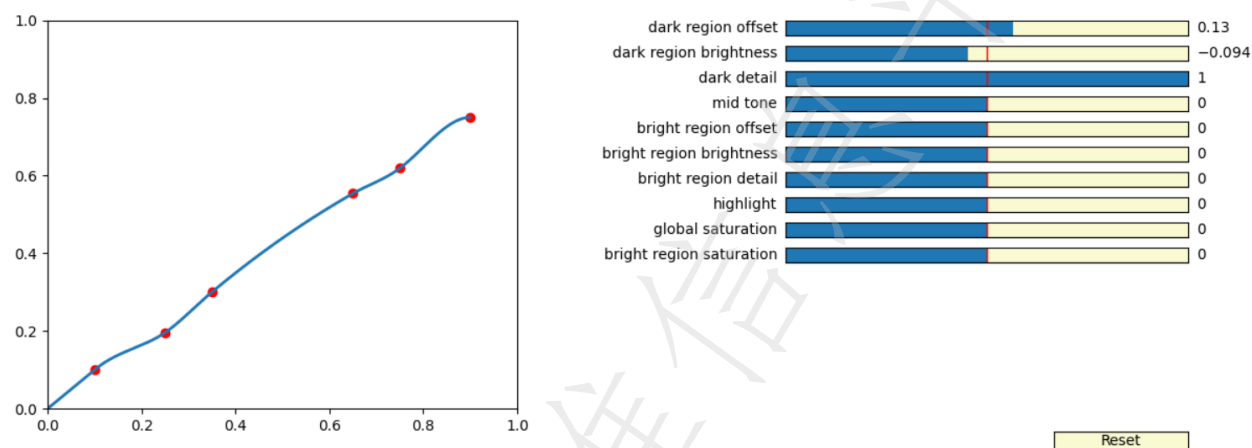


图4 HDR Vivid导演模式界面参考

## 12.3.2.2.导演模式组成

导演模式的规范主要包括两部分。第一,运行导演模式前,必须先运行自动模式。自动模式提取的动态元数据,和对应的色调映射曲线,应当作为导演模式的参照。假如调色师没有做出任何调节,那么导演模式输出应当等同于自动模式的输出。或者,调色师在调节过程中,恢复默认设置,那么动态元数据和色调映射曲线需要回到自动模式的状态。调色师调节完毕之后,用调色师调节过的动态元数据替代自动模式的动态元数据,并写入到输出文件。第二,控件的数量、调节范围和调节的功能要符合HDR Vivid的规范。HDR Vivid建议10个控件,类型为滑块,每个控件的调节范围都在-1.0和1.0之间,默认值是0,对应自动模式。当所有控件的值为0时,动态元数据和色调映射曲线要和自动模式一致。

## 12.3.2.3.曲线锚点

HDR Vivid曲线由6个锚点决定,建议在界面中画出这6个锚点的位置。而导演模式的控件,直接在调节这6个锚点的位置,并根据调整来计算HDR Vivid曲线并生成对应的动态元数据。这6个锚点的x值从小到大依次就是HDR Vivid动态元数据里定义的TH1、TH2、TH3、TH1\_HIGH、TH2\_HIGH、TH3\_HIGH。纵坐标就是HDR Vivid曲线在这六个位置的y值,表示为TH1\_Y、TH2\_Y、TH3\_Y、TH1\_HIGH\_Y、TH2\_HIGH\_Y、TH3\_HIGH\_Y。导演模式以自动模式为基础,自动模式对应的锚点值表示为TH1\_ref、TH2\_ref、TH3\_ref、TH1\_HIGH\_ref、TH2\_HIGH\_ref、TH3\_HIGH\_ref、TH1\_Y\_ref、TH2\_Y\_ref、TH3\_Y\_ref、TH1\_HIGH\_Y\_ref、TH2\_HIGH\_Y\_ref、TH3\_HIGH\_Y\_ref。

曲线调节算法负责把控件的值映射到锚点的位置,进而计算出曲线参数和动态元数据。6个锚点的位置是相互依赖的,曲线调节算法会最大程度的协调6个锚点的位置,力求生成符合HDR Vivid标准的曲线。但是,在某些极端情况下,不存在一条符合HDR Vivid标准的曲线通过调节的6个锚点位置,曲线调节算法会反馈拟合失败,这种情况下,需要控件、曲线和动态元数据返回到上一次的正常状态。

## 12.3.2.4.控件

各控件的功能和对应的锚点位置以及动态元数据如表16所述。

表16 控件描述

编号	控件	描述
1	暗区偏移	<p>调节的是TH3的值，对应动态元数据里定义的TH3。</p> <p>控件值为0时，TH3的值等于自动模式下的TH3_ref；-1时，TH3取调节范围的最小值；+1时，取调节范围的最大值。调节范围建议采用[TH2_ref,0.45]。</p> <p>小于TH3的区域被定义为暗区。TH1、TH2、TH3这三个锚点确定暗区的三次样条曲线。调节TH3的值会改变暗区三次样条的覆盖范围。另外，主曲线是由TH3和TH1_HIGH这两个锚点确定的，所以改变TH3,需要重新计算主曲线的参数，包括动态元数据里定义的m_p, m_a, m_b, m_m, m_n, K1, K2, K3。</p>
2	暗区亮度	<p>调节的是TH1_Y的值，对应动态元数据里定义的MB的值，也就是曲线开始处一次样条的斜率，会影响图像最暗部分的亮度。</p> <p>控件取值为0时，MB取自动模式下值MB_ref；-1时，取调节范围的最小值；+1时，最大值。调节范围建议[0,1]。</p>
3	暗部细节	<p>调节的是TH2_Y的值，对应动态元数据里定义的strength，影响暗区三次样条曲线的形状，进而影响暗区的细节表现。</p> <p>控件值取0时，strength取自动模式下的值strength_ref；-1时，取调节范围的最小值；+1时，最大值。建议的取值范围是[-0.5,0.5]。</p>
4	中灰亮度	<p>调节的是TH3_Y的值。TH3_Y定义为中灰亮度，没有直接对应的动态元参数，但是会影响到动态元数据里定义的m_p, m_a, m_b, m_m, m_n, K1, K2, K3。</p> <p>控件值取0时，TH3_Y取自动模式下的值TH3_Y_ref；-1时，取调节范围的最小值；+1时，最大值。调节范围建议采用[TH2_Y_ref,TH1_HIGH_Y_ref]。</p> <p>改变TH3_Y的值，需要重新计算主曲线参数，包括动态元数据里定义的m_p, m_a, m_b, m_m, m_n, K1, K2, K3。假如TH3_Y调到某一位置以后，计算不出合规的HDR Vivid曲线，那么触发联动机制，即在TH3_Y调节的相同方向上，搜索新的TH1_HIGH_Y的值，直到可以计算出合规的HDR Vivid曲线。假如失败了，则说明当前调节位置不合法，退回上一个合法的状态和控件值。</p>
5	亮区偏移	<p>调节的是TH1_HIGH的值，对应动态元数据里定义的TH1_HIGH。</p> <p>控件值为0时，TH1_HIGH的值等于自动模式下的TH1_HIGH_ref；-1时，TH1_HIGH取调节范围的最小值；+1时，取调节范围的最大值。调节范围建议采用[0.48,TH2_HIGH_ref]。</p> <p>大于TH1_HIGH的部分定义为亮区。TH1_HIGH、TH2_HIGH、TH3_HIGH这三个锚点确定暗区的三次样条曲线。调节TH1_HIGH的值会改变亮区三次样条的覆盖范围。另外，主曲线是由TH3和TH1_HIGH这两个锚点确定的，所以改变TH1_HIGH,需要重新计算主曲线的参数，包括动态元数据里定义的m_p, m_a, m_b, m_m, m_n, K1, K2, K3。</p>
6	亮区亮度	<p>调节的是TH1_HIGH_Y的值，没有直接对应的动态元参数，但是会影响到动态元数据里定义的m_p, m_a, m_b, m_m, m_n, K1, K2, K3。</p> <p>控件值取0时，TH1_HIGH_Y取自动模式下的值TH1_HIGH_Y_ref；-1时，取调节范围的最小值；+1时，最大值。调节范围建议采用[TH3_Y_ref,TH2_HIGH_Y_ref]。</p> <p>改变TH1_HIGH_Y的值，需要重新计算主曲线参数，包括动态元数据里定义的m_p, m_a, m_b, m_m, m_n, K1, K2, K3。假如TH1_HIGH_Y调到某一位置以后，计算不出合规的HDR Vivid曲线，那么触发联动机制，即在TH1_HIGH_Y调节的相同方向上，搜索新的TH3_Y的值，直到可以计算出合规的HDR Vivid曲线。假如失败了，则说明当前调节位置不合法，退</p>

编号	控件	描述
		回上一个合法的状态和控件值。
7	亮区 细节	调节的是TH2_HIGH_Y的值，对应动态元数据里定义的strength_HIGH，影响亮区三次样条曲线的形状，进而影响亮区的细节表现。 控件值取0时，strength_HIGH取自动模式下的值strength_HIGH_ref；-1时，取调节范围的最小值；+1时，最大值。建议的取值范围是[-0.5,0.5]。
8	高光	调节的是TH3_HIGH的值，对应动态元数据里定义的TH3_HIGH。 控件值取0时，TH3_HIGH等于自动模式的值TH3_HIGH_ref；-1时，取调节范围的最小值；+1时，最大值。建议的取值范围是[TH2_HIGH,max_source],max_source表示当前帧的最大亮度的PQ值。 TH3_HIGH_Y在导演模式下，一定等于参考显示器的最大亮度，所以大于TH3_HIGH的区域属于过曝区域。这个控件就是用来调节过曝区域的范围，以控制高光的效果。
9	整体 饱和度	调节的是整个亮度范围的饱和度，对应动态元参数里定义的color_saturation_gain[0]。 控件取值为0时，对应自动模式下color_saturation_gain[0]的值；-1时，取最小值0；+1时，最大值255。
10	亮区 饱和度	针对高亮区域调节饱和度，对应动态元参数里定义的color_saturation_gain[1]。 控件取值为0时，对应自动模式下color_saturation_gain[1]的值；-1时，取最小值0；+1时，最大值255。

### 13.内容保护

HDR Vivid的视频流可以被正常的数字版权保护系统加密和解密。因其关联的元数据并不能还原或者反向推导出视频流，所以不建议对HDR的元数据进行额外的内容保护，增加系统复杂度。